







**Содержимое ASM-файла:**

;Задание No1. Написать программу, которая определяет в данном слове

;(см. таблицу No1) содержимое разрядов i и i + 1 и формирует число К в виде:

;0, если i = 0 и i + 1 = 0

;К = 1, если i = 0 и i + 1 = 1

;2, если i = 1 и i + 1 = 0

;3, если i = 1 и i + 1 = 1,

;где i – номер варианта.

Chinik segment para 'code'

assume cs:Chinik,ds:Chinik,ss:Chinik,es:Chinik

org 100h ; пропускаем первые 256 байт (.com)

begin: jmp main

; это мои данные (переменные)

;---------------------------------

date dw ?

my\_s db '+'

T\_Th db ?

Th db ?

Hu db ?

Tens db ?

Ones db ?

;---------------------------------

A dw 068BCh

OTV dw ?

MASKA dw 0000000000011000b

main proc near

mov ax,A

and ax,MASKA

; первый вариант

cmp ax,0

jz m1

; второй вариант

cmp ax,10h

jz m2

cmp ax,8h

jz m3

mov OTV, 3 ; 0000 0000 0001 1000

jmp konec

m1: ; результат 0000 0000 0000 0000

mov OTV,0

jmp konec

m2: ; результат 0000 0000 0001 0000

mov OTV,1

jmp konec

m3: ; результат 0000 0000 0000 1000

mov OTV,2

konec: mov ax,OTV

mov date,ax

call disp

ret

main endp

;------------------------------------------

; Процедура выводит результат вычислений, помещенный в data

DISP proc near

;----- Вывод результата на экран ----------------

;--- Число отрицательное ?----------

mov ax,date

and ax,1000000000000000b

mov cl,15

shr ax,cl

cmp ax,1

jne @m1

mov ax,date

neg ax

mov my\_s,'-'

jmp @m2

;--- Получаем десятки тысяч ---------------

@m1: mov ax,date

@m2: cwd

mov bx,10000

idiv bx

mov T\_Th,al

;------- Получаем тысячи ------------------------------

mov ax,dx

cwd

mov bx,1000

idiv bx

mov Th,al

;------ Получаем сотни ---------------

mov ax,dx

mov bl,100

idiv bl

mov Hu,al

;---- Получаем десятки и единицы ----------------------

mov al,ah

cbw

mov bl,10

idiv bl

mov Tens,al

mov Ones,ah

;--- Выводим знак -----------------------

cmp my\_s,'+'

je @m500

mov ah,02h

mov dl,my\_s

int 21h

;---------- Выводим цифры -----------------

@m500: cmp T\_TH,0 ; проверка на ноль

je @m200

mov ah,02h ; выводим на экран, если не ноль

mov dl,T\_Th

add dl,48

int 21h

@m200: cmp T\_Th,0

jne @m300

cmp Th,0

je @m400

@m300: mov ah,02h

mov dl,Th

add dl,48

int 21h

@m400: cmp T\_TH,0

jne @m600

cmp Th,0

jne @m600

cmp hu,0

je @m700

@m600: mov ah,02h

mov dl,Hu

add dl,48

int 21h

@m700: cmp T\_TH,0

jne @m900

cmp Th,0

jne @m900

cmp Hu,0

jne @m900

cmp Tens,0

je @m950

@m900: mov ah,02h

mov dl,Tens

add dl,48

int 21h

@m950: mov ah,02h

mov dl,Ones

add dl,48

int 21h

mov ah,02h

mov dl,10

int 21h

mov ah,02h

mov dl,13

int 21h

;-------------------------------------

mov ah,08

int 21h

ret

DISP endp

Chinik ends

end begin

Содержимое файла 2:

;Вариант 4

;Задание No2. Написать программу, основная часть которой выполняет

;следующие действия:

;1) передает через стек параметры в первую процедуру (см.номер варианта

;в таблице No2).

;2) вычисляет значение выражения (см.номер варианта в таблице No3) и

;передает его во вторую процедуру.

;3) первая процедура возвращает в основную программу значение

;переменной Y, вычисляющееся по формуле, приведенной ниже (см. номер

;варианта в таблице No2).

;4) вторая процедура возвращает в основную программу значение

;переменной Y2, вычисляющееся по формуле, приведенной ниже (см. номер

;варианта в таблице No3).

;P.S. Значения параметров A, A1, A2, B, X, X1, X2, X3 выбираются произвольно.

;4. Передаваемые параметры: Z1 = A + B, Z2 = A – B, Z3 = –(A + B).

;Возвращаемое значение:

;Y = 0, если Z1 >= 1

;Y = 1, если Z2 >= 1

;Y = 2, если Z3 >= 1

;Y = 3, в противном случае

;4. Функции вызывающей программы:

;а) Y1=(A1+A2)/10

;б) Поместить в ячейку памяти Y2 возвращенное из подпрограммы значение

;Функции подпрограммы:

;Y2 = FF, если Y1 > 0

;Y2 = DD, если Y1 <= 0

Chinik segment para 'code'

assume cs:Chinik,ds:Chinik,ss:Chinik,es:Chinik

org 100h ; пропускаем первые 256 байт (.com)

begin: jmp main

; это мои данные (переменные)

;---------------------------------

date dw ?

my\_s db '+'

T\_Th db ?

Th db ?

Hu db ?

Tens db ?

Ones db ?

Y dw ?

Z1 dw ?

Z2 dw ?

Z3 dw ?

A1 dw 5

A2 dw 10

Y1 dw ?

Y2 dw ?

;---------------------------------

A dw 400

B dw 400

adres dw ?

main proc near

    mov ax,A

    add ax,B

    mov Z1,ax ; Z1 = ax=A+B

    ;

    mov ax,A

    sub ax,B

    mov Z2,ax ; Z2 = A – B

    ;

    mov ax,0

    sub ax,A

    sub ax,B

    mov Z3,ax ; Z3 = –(A + B).

    push Z1 ; в верхушке стека Z1 = 800

    push Z2 ; в верхушке стека Z2 = 0

    push Z3 ; в верхушке стека Z3 = –800

    call myproc ; в верхушке стека адрес возврата

    pop ax ; достаем ответ из стека (стек теперь пустой)

    mov date,ax

    call disp

    mov ax,A1

    add ax,A2

    cwd

    mov bx,10

    idiv bx

    mov Y1,ax

    push Y1

    call myproc2 ; положили в стек адрес возврата

    pop ax ; достаем ответ из стека

    mov date,ax

    call disp

    ret

main endp

;------------------------------------------

myproc proc near

    pop adres ; достаем из стека адрес возврата в ячейку adres

    pop Z1 ; Z1=800

    pop Z2 ; Z2 = 0

    pop Z3 ; Z3 = –800 стек пустой

    cmp Z1,1

    JB m1 ; Z1 меньше единицы, уходим на метку m1

    mov Y,0 ; Z1 больше единицы, тогда Y = 0

    jmp konec

    m1: ; Z1 меньше единицы

    cmp Z2,1

    JB m2

    mov Y,1

    jmp konec

    m2: ; Z2 меньше единицы

    cmp Z3,1

    JB m3

    mov Y,2

    jmp konec

    m3: mov Y,3 ; Z3 меньше единицы

    konec: push Y ; кладем в стек Y (ответ)

    push adres ; возвращаем на верхушку стека адрес возврата

    ret

myproc endp

;------------------------------------------

myproc2 proc near

    pop adres

    pop Y1

    cmp Y1,0

    jbe z ; если Y1<=0, то уходим на метку Z

                ;если Y1>0

    mov Y2,00FFh ; заносим в ячейку "FF" (255)

    jmp konec1

    z: ; если Y1<=0

    mov Y2,00DDh ; заносим в ячейку "DD" (221)

    konec1: push Y2

    push adres

    ret

myproc2 endp

DISP proc near

;----- Вывод результата на экран ----------------

;--- Число отрицательное ?----------

mov ax,date

and ax,1000000000000000b

mov cl,15

shr ax,cl

cmp ax,1

jne @m1

mov ax,date

neg ax

mov my\_s,'-'

jmp @m2

;--- Получаем десятки тысяч ---------------

@m1: mov ax,date

@m2: cwd

mov bx,10000

idiv bx

mov T\_Th,al

;------- Получаем тысячи ------------------------------

mov ax,dx

cwd

mov bx,1000

idiv bx

mov Th,al

;------ Получаем сотни ---------------

mov ax,dx

mov bl,100

idiv bl

mov Hu,al

;---- Получаем десятки и единицы ----------------------

mov al,ah

cbw

mov bl,10

idiv bl

mov Tens,al

mov Ones,ah

;--- Выводим знак -----------------------

cmp my\_s,'+'

je @m500

mov ah,02h

mov dl,my\_s

int 21h

;---------- Выводим цифры -----------------

@m500: cmp T\_TH,0 ; проверка на ноль

je @m200

mov ah,02h ; выводим на экран, если не ноль

mov dl,T\_Th

add dl,48

int 21h

@m200: cmp T\_Th,0

jne @m300

cmp Th,0

je @m400

@m300: mov ah,02h

mov dl,Th

add dl,48

int 21h

@m400: cmp T\_TH,0

jne @m600

cmp Th,0

jne @m600

cmp hu,0

je @m700

@m600: mov ah,02h

mov dl,Hu

add dl,48

int 21h

@m700: cmp T\_TH,0

jne @m900

cmp Th,0

jne @m900

cmp Hu,0

jne @m900

cmp Tens,0

je @m950

@m900: mov ah,02h

mov dl,Tens

add dl,48

int 21h

@m950: mov ah,02h

mov dl,Ones

add dl,48

int 21h

mov ah,02h

mov dl,10

int 21h

mov ah,02h

mov dl,13

int 21h

;-------------------------------------

mov ah,08

int 21h

ret

DISP endp

Chinik ends

end begin

Работа программы:

